

آموزش سی شارپ (C#) قسمت ۱۴ : آرایه ها و دستور foreach (نسخه PDF)

در بخش قبلی آموزش زبان سی شارپ با دستورات کنترلی و حلقه ها آشنا شدیم. در این بخش قصد داریم با مبحث آرایه ها و دستور foreach آشنا شویم. همانطور که در قسمت آشنایی با متغیرها گفتیم، یک متغیر خانه ای از حافظه است که می توان مقداری را در داخل آن ذخیره کرد. برای مثال دستور زیر متغیری از نوع string و با نام name تعریف کرده و مقدار آنرا داخل آن قرار می دهد:

```
string name = "ITPro";
```

اما فرض کنید که بخواهیم در یک متغیر، بیش از یک مقدار را ذخیره کنیم. برای مثال، ما می خواهیم لیستی از نمرات یک دانشجو را در برنامه داشته باشیم. برای اینکار باید متغیری تعریف کنیم که قابلیت نگهداری بیش از یک مقدار را داشته باشد. در اینجا مبحث آرایه ها مطرح می شود. آرایه ها این قابلیت را به ما می دهند که لیستی از یک نوع را نگهداری کنیم. شیوه تعریف آرایه ها در زبان سی شارپ به صورت زیر است:

```
{type}[] numbers = new {type}[length];
```

در قسمت type نوع داده ای که قصد داریم به عنوان آرایه ایجاد شود و در قسمت length باید تعداد خانه های آرایه را مشخص کنیم. برای مثال فرض کنید که قصد داریم یک آرایه ۱۰ خانه ای از نوع int ایجاد کنیم:

```
int[] numbers = new int[10];
```

بوسیله کد بالا یک آرایه ۱۰ خانه ای از نوع int برای ما ایجاد می شود. برای مثال دوم یک آرایه ۲۰ خانه ای از نوع string ایجاد می کنیم:

```
string[] names = new string[20];
```

در مرحله بعد باید با نحوه دسترسی به خانه های آرایه آشنا شویم. در دو حالت ما نیاز به دسترسی به خانه های آرایه خواهیم داشت:

۱. قرار دادن مقدار داخل یکی از خانه های آرایه

۲. گرفتن مقدار یکی از خانه های آرایه

ابتدا نحوه قرار دادن مقدار را با هم بررسی می کنیم:

```
string[] names = new string[20];  
names[10] = "ITPro.ir";
```

همانطور که مشاهده می کنید ابتدا نام متغیر، سپس داخل [] اندیس خانه ای از آرایه که قصد داریم مقدار داخل آن قرار بگیرد را می نویسیم. اما به یک نکته توجه کنید که اندیس آرایه ها از شماره ۰ شروع می شود. یعنی در کد بالا که عدد ۱۰ را نوشتیم، اشاره به خانه نهم آرایه می کنید. برای مثال در یک آرایه ۱۰ خانه ای، عدد ۰ به خانه اول و عدد ۹ به خانه دهم اشاره می کند.

برای خواندن مقدار یکی از خانه های آرایه به صورت زیر عمل می کنیم:

```
string[] names = new string[10];  
names[0] = "Reza";  
names[1] = "Hossien";  
names[2] = "Mohammad";
```

```
Console.WriteLine(names[1]); // output Hossein
```

در زبان سی شارپ به صورت های مختلفی می توان آرایه را تعریف و مقدار دهی کرد، یکی از حالت ها را بررسی کردیم. نوع دیگری از تعریف آرایه به این صورت است که ما هنگام تعریف آرایه مقادیر خانه های آن را مشخص می کنیم و بر اساس مقادیر، تعداد خانه های آرایه تعیین می شود:

```
int[] numbers = new[] { 2,6,3,1,7,9};
```

در کد بالا، آرایه ای از نوع int تعریف شده و مقادیر مشخص شده داخل {} در خانه های آرایه قرار میگیرد. در مثال قبلی آرایه تعریف شده حاوی ۶ خانه خواهد بود (بر اساس تعداد اعداد داخل {}).

می توان برای راحتی کار، از نوشتن int[] صرفنظر کرد:

```
int[] numbers = { 2,6,3,1,7,9};
```

کد بالا، دقیقاً کار کد قبلی را انجام می دهد.

زمانی که ما یک آرایه تعریف می کنیم، می توانیم بوسیله دستور for به خانه های آرایه دسترسی داشته باشیم. برای مثال، فرض کنید برنامه ای می خواهیم بنویسیم که ۱۰ عدد را از کاربر دریافت کرده، سپس حاصل جمع این ۱۰ عدد را در خروجی چاپ کند.

۱. در ابتدا باید آرایه ای که اعداد داخل آن باید ذخیره شوند را تعریف کنیم:

```
int[] nums = new int[10];
```

۲. در قدم بعدی باید ۱۰ عدد را از کاربر دریافت کرده و در خانه های آرایه ذخیره کنیم. دقت کنید، ۱۰ مرتبه عملیات دریافت اطلاعات از کاربر تکرار می شود، پس ما باید یک حلقه تشکیل داده و داخل آن حلقه اقدام به دریافت اعداد از کاربر نماییم:

```
for (int index = 0; index <= 9; index++)
{
    Console.WriteLine("Enter number " + (index + 1) + " of 10: ");
    nums[index] = int.Parse(Console.ReadLine());
}
```

دقت کنید که شمارنده حلقه از عدد صفر شروع شده و تا زمانی که مقدار شمارنده کوچکتر یا مساوی عدد ۹ باشد تکرار خواهد شد. ما می توانیم از عدد شمارنده برای دسترسی به خانه های آرایه نیز استفاده کنیم. همچنین قبل از دریافت عدد، پیغامی مناسب به کاربر جهت وارد کردن عدد نمایش می دهیم.

۳. در مرحله بعد، باید تک تک خانه های آرایه را خوانده و حاصل جمع را در یک متغیر جانبی ذخیره کنیم:

```
int sum = 0;
for (int index = 0; index <= 9; index++)
{
    sum = sum + nums[index];
}
Console.WriteLine("Sum of entered numbers: " + sum);
```

می توان دستور داخل حلقه for را به صورت خلاصه تر نوشت:

```
int sum = 0;
for (int index = 0; index <= 9; index++)
{
    sum += nums[index];
}
Console.WriteLine("Sum of entered numbers: " + sum);
```

خاصیت Length

فرض کنید، ما طول آرایه ای که تعریف کردیم را نمی دانیم! در این مواقع می توان با خاصیت Length طول آرایه را بدست آورد. در مثال زیر نحوه بدست آوردن طول آرایه با خاصیت Length را بررسی می کنیم:

```
string[] names = {"ITPro.IR", "Mohammad", "Hossein", "Farhad", "Saeed", "Sadegh", "Mahmoud"};
Console.WriteLine(names.Length); // Output => 7
```

دستور بالا، مقدار ۷ که تعداد آیتم های آرایه یا طول آرایه می باشد را در خروجی چاپ می کند. این خاصیت مقداری از نوع int برای ما بر می گرداند.

حال مثال قبلی که حاصل جمع اعداد را حساب می کرد به صورت زیر تغییر می دهیم:

```
int sum = 0;
for (int index = 0; index < nums.Length; index++)
{
    sum += nums[index];
}
Console.WriteLine("Sum of entered numbers: " + sum);
```

ما قسمت شرط دستور for را جوری تغییر دادیم که حلقه تا زمانی که شمارنده کوچکتر از طول آرایه باشد تکرار شود. دقت کنید که به جای \leq ما از $<$ استفاده کردیم. زیرا Length طول آرایه را برمیگرداند و به دلیل شروع شدن خانه های آرایه از اندیس ۰ حلقه باید یک عدد کمتر از طول خانه های آرایه تکرار شود.

آرایه های چند بعدی

در مثال های قبلی، آرایه هایی که ایجاد کردیم، متغییرهایی بودند که شامل چندین خانه بودند. به این آرایه ها آرایه های خطی نیز می گویند. اما فرض کنید قصد داریم جدول ضرب ۹ در ۹ را در یک آرایه ذخیره کنیم. برای این کار باید از آرایه های چند بعدی استفاده کنیم. برای مثال، دستور زیر یک آرایه ۲ بعدی شامل ۹ سطر و ۹ ستون را تعریف می کند:

```
int[,] matrix = new int[9,9];
```

برای ابعاد بیشتر باید تعداد اعداد داخل [] را بیشتر کنیم:

```
int[,,,] matrix = new int[9,9,9];
```

در مثال زیر، یک آرایه ۹ در ۹ تعریف کرده و جدول ضرب ۹ در ۹ را داخل آن ذخیره می کنیم و در نهایت خروجی را چاپ می کنیم:

```
int[,] matrix = new int[9,9];

for (int col = 0; col < 9; col++)
{
    for (int row = 0; row < 9; row++)
    {
        matrix[col, row] = (col + 1)*(row + 1);
    }
}

for (int col = 0; col < 9; col++)
{
    for (int row = 0; row < 9; row++)
    {
        Console.Write(matrix[col, row] + "\t");
    }
    Console.WriteLine();
}
```

آرایه های نامنظم یا Jagged Arrays

در قسمت قبلی با آرایه های چند بعدی آشنا شدیم، مثل آرایه های ۲*۲ یا ۳*۳ یا ۴*۲، اما فرض کنید قصد داریم آرایه دو بعدی با تعداد خانه های نامساوی تعریف کنیم، در قسمت آرایه های چند بعدی هر سطر از آرایه دارای تعداد خانه های مشخص بود، مثل ۳ سطر و ۲ ستون، اما فرض کنید می خواهیم هر سطر از آرایه تعداد خانه های متفاوتی داشته باشد. برای مثال، سطر اول ۵ خانه، سطر دوم ۶ خانه، سطر سوم ۲ خانه و ... برای اینجور مواقع ما می توانیم از آرایه های نامنظم استفاده کنیم. شیوه تعریف آرایه های نامنظم به صورت زیر است:

```
int[][] jaggedArray = new int[3][];
```

با دستور بالا یک آرایه سه خانه ای که هر یک از خانه های آن نیز یک آرایه است تعریف کردیم. حالا باید برای تک تک خانه های آرایه نیز، آرایه ایجاد کنیم:

```
jaggedArray[0] = new int[5];
jaggedArray[1] = new int[4];
jaggedArray[2] = new int[2];
```

می توان هر یک از خانه های آرایه را به صورت مستقیم مانند مثال زیر مقدار دهی نیز کرد:

```
jaggedArray[0] = new int[] { 1, 3, 5, 7, 9 };
jaggedArray[1] = new int[] { 0, 2, 4, 6 };
jaggedArray[2] = new int[] { 11, 22 };
```

در مثال بالا، یک آرایه ۳ خانه ای که خانه اول دارای ۵ خانه، خانه دوم دارای سه ۳ و خانه سوم دارای ۲ خانه می باشد را ایجاد کردیم.

برای دسترسی به خانه های آرایه باید از دو اندیس استفاده کرد، اندیس اول برای آرایه اول و اندیس دوم برای آرایه دوم:

```
jaggedArray[0][1] = 77;  
Console.WriteLine("{0}", jaggedArray[0][1]);
```

دستور foreach

در زبان سی شارپ، زمانی که ما یک آرایه تعریف می کنیم، در حقیقت یک مجموعه قابل شمارش (Enumerable) تعریف کرده ایم. در دات نت مجموعه های قابل شمارش زیادی وجود دارند که در بخش های بعدی با آنها بیشتر آشنا خواهیم شد. اما منظور از قابل شمارش چیست؟ نوع های قابل شمارش، نوع هایی می باشند که ما می توانیم بوسیله دستور foreach به خانه های آنها دسترسی داشته باشیم. ساختار کلی دستور foreach به صورت زیر است:

```
foreach({type} {variable-name} in {collection})  
{  
    // loop syntaxes  
}
```

۱. قسمت type باید نوع متغیری که مجموعه بر اساس آن تعریف شده را مشخص کنیم. برای مثال، اگر آرایه ای داریم که خانه های آن از نوع int است، باید به جای type نوع int را بنویسیم. به جای type می توان از کلمه کلیدی var که در بخش های قبلی با آن آشنا شدیم نیز استفاده کرد.

۲. قسمت variable-name باید نام متغیری که خانه های آرایه در آن قرار می گیرند را مشخص کنیم. این نام کاملاً دلخواه است.

۳. در قسمت collection نام مجموعه ای که قصد داریم به خانه های آن دسترسی داشته باشیم را می نویسیم.

شیوه عملکرد دستور foreach به این صورت است که حلقه ای بر روی مجموعه مورد نظر ایجاد می شود، سپس هر یک از خانه های آن مجموعه خوانده شده و مقدار آن داخل متغیری که نام آن را در بخش variable-name مشخص کردیم قرار میگیرد و ما می توانیم در هر تکرار به آیتم خانه های آرایه به ترتیب قرار گیری در مجموعه دسترسی داشته باشیم. به مثال زیر توجه کنید:

```
string[] names = {"ITPro.IR", "Mohammad", "Hossein", "Farhad", "Saeed", "Sadegh", "Mahmoud"};  
  
foreach (var name in names)  
{  
    Console.WriteLine(name);  
}
```

بوسیله دستور بالا، کلیه آیتم های آرایه names در خروجی چاپ خواهد شد. دستور foreach بالا دقیقاً معادل دستور for زیر می باشد:

```
for (int index = 0; index < names.Length; index++)  
{  
    Console.WriteLine(names[index]);  
}
```

دستور foreach هنگام کار با مجموعه ها به شدت مورد استفاده قرار می گیرد. اما توجه داشته باشید اگر در برنامه ای که قصد نوشتن آن را

دارید، سرعت اجرای کدها از اهمیت زیادی برخوردار است، دستور for سرعت اجرای بیشتری نسبت به دستور foreach دارد.

با اتمام بخش آرایه ها، در بخش بعدی در مورد متدها و کاربرد متدها در برنامه نویسی توضیح خواهیم داد. امیدوارم که در طول این دوره آموزشی با بنده و وب سایت ITPRO همراه باشید. ITPRO باشید

نویسنده : حسین احمدی

منبع : جزیره برنامه نویسی وب سایت توسینسو

هرگونه نشر و کپی برداری بدون ذکر منبع و نام نویسنده دارای اشکال اخلاقی است

#آرایه ها در سی شارپ #استفاده از دستور foreach در سی شارپ #آموزش زبان سی شارپ
#آموزش برنامه نویسی C# #آموزش مقدماتی برنامه نویسی #آشنایی با jagged_arrays #دستور foreach
#آموزش برنامه نویسی #آرایه های نامنظم در سی شارپ

azad1000

من میخام به یک آرایه دو بعدی مقدار دهی کنم که دو بعد اون از یک نوع نیستند.
مثلا یک بعد اسامی دانشجویان و یک بعد معدل اونهاست و بعد میخام دانشجویها رو طبق معدلشون کلاس بندی کنم.
چطور باید مقداردهی رو انجام بدم؟

فرهاد خانلری

عالی بود مهندس جان جهت درک بیشتر و تمرین نظر شما چیه که انتهای هر درس یا هر دوره یک یا دو مثال قرار دهید ، مثال هایی که تمام اون بخش یا دوره رو در بر بگیره و دوستان خودشون جواب بدن و در نهایت شما جواب رو انتخاب کنی یا خودت جواب رو بدی.
اینجوری هم همکاری بین دوستان زیاد خواهد شد هم حس یادگیری برانگیخته خواهد شد.

mdg2000

مهندس جان چرا ادامه نمیدی؟

فرهاد خانلری

کاربر mdg2000 مهندس حسین احمدی Developer سایت itpro هستند و این روز ها سخت مشغول به روز رسانی نرم افزار سایتمون هستند و سرشون شلوغه .
شکيبا باشید.

mhrnia

سلام

اولین باره دارم با برنامه سی شارپ کار میکنم (خواهشا راهنمایی کنید)
برنامه ای طول عرض مستطیلی و مربعی را خوانده محیط و مساحت مستطیل و مربع رامحاسبه می کند ؟؟؟؟؟
آیا برای مستطیل و مربع برنامه جداگانه ای بنوسیم ؟؟؟

حسین احمدی

سلام دوست عزیز، لطفاً سوالتون رو در قالب یک سوال مطرح کنید.

mhrnia

انجام دادم نشد همیشه راهنمایی کنید!!

maverick

سلام .

میشه قطعه کد مقایسه دو آرایه یک بعدی که هر کدام ۵ خانه دارند رو بنویسید

یعنی تک تک خانه ها رو با هم مقایسه کنه و هر دو از نوع int هستند

یعنی اگه مثلا خانه ۱ اولی با خانه ۳ دومی برابر بود یه عمل دیگه انجام بده و اگه مثلا ۲ اولی با ۲ دومی برابر بود یه کار دیگه انجام بده.

محمد محرابی

چند مثال

مثال ۱. در مثال زیر ابتدا آرایه ای با استفاده از حلقه for با مقادیری پر میشود؛ سپس با استفاده از حلقه foreach مقادیر داخل آرایه در خروجی چاپ میشود.

```
int sum = 0;
int[] nums = new int[10]; // تعریف آرایه با 10 خانه

// پرکردن آرایه
for (int i = 0; i < 10; i++)
    nums[i] = i;

Console.WriteLine("use foreach to display and sum the values");

// چاپ مقادیر داخل آرایه در خروجی
foreach (int x in nums)
{
    Console.WriteLine("Value is: " + x);
    sum += x; // در هر دور از حلقه مقدار هر خانه از آرایه با مقادیر قبل جمع میشود
}

Console.WriteLine("Summation: " + sum); // نمایش مجموع مقادیر داخل آرایه
```

خروجی

use foreach to display and sum the values

Value is: 0

Value is: 1

Value is: 2

```
Value is: 3
Value is: 4
Value is: 5
Value is: 6
Value is: 7
Value is: 8
Value is: 9
Summation: 45
```

مثال ۲. یک آرایه دوبعدی تعریف میشود سپس با استفاده از دو حلقه از مقادیر پر میشود. سپس با استفاده از حلقه foreach مقدار هر درایه از ماتریس (آرایه ۳x۵) در خروجی چاپ میشود. مجموع درایه ها نیز محاسبه و چاپ میشود.

```
int sum = 0;
int[,] nums = new int[3,5];

for(int i = 0; i < 3; i++)
    for(int j=0; j < 5; j++)
        nums[i,j] = (i+1)*(j+1);

foreach(int x in nums) {
    Console.WriteLine("Value is: " + x);
    sum += x;
}

Console.WriteLine("Summation: " + sum);
```

خروجی

```
Value is: 1
Value is: 2
Value is: 3
Value is: 4
Value is: 5
Value is: 2
Value is: 4
Value is: 6
Value is: 8
Value is: 10
Value is: 3
Value is: 6
Value is: 9
Value is: 12
Value is: 15
Summation: 90
```


مثال ۳. جستجو در آرایه با استفاده از حلقه foreach.

در مثال زیر مقدار ۵ در آرایه سرچ میشود و در صورتی که یافت شد متغیر found که با مقدار false از قبل پر شده بود با مقدار جدید true مقدار دهی میشود و با بررسی شرط و انجام دستور داخل شرط (break) از بدنه حلقه خارج میشود. سپس چون مقدار متغیر true است با شرط آخر برنامه مقدار Value found! در خروجی چاپ خواهد شد.

```
int[] nums = new int[10];
int val;
bool found = false;

for(int i = 0; i < 10; i++)
    nums[i] = i;

val = 5;

foreach(int x in nums) {
    if(x == val) {
        found = true;
        break;
    }
}

if(found)
    Console.WriteLine("Value found!");
```

خروجی

Value found!

hoseinsalime

خروجی روی تکست باکس-یا روی فرم چاپ شه

```
int[,] matrix = new int[9,9]
for (int col = 0; col < 9; col++)
{
    for (int row = 0; row < 9; row++)
    {
        matrix[col, row] = (col + 1)*(row + 1);
    }
}
```

```
{
    (++for (int col = 0; col < 9; col
    }
    (++for (int row = 0; row < 9; row
    }
    ;("Console.WriteLine(matrix[col, row] + "\t
    {
    ;()Console.WriteLine
    {
```

zahra .m

سلام استاد.وقت بخیر

در آرایه های نامنظم میشه از دو نوع داده مختلف برای هر گروه آرایه استفاده کرد؟

حسین احمدی

سلام، برای آرایه های نامنظم که هر خونه شامل یک نوع داده باشه شما باید آرایه رو از نوع object تعریف کنید:

```
var objs = new object[10];
objs[0] = new int[] { 4, 5, 7, 1 };
objs[1] = new string[] { "Item1", "Item2" };
```

zahra .m

متشکرم استاد!

مطلب اصلی